一种局部二值模式图像特征点匹配算法 *

王　强[1,2]，李柏林[1]，罗建桥[1]，陈小艳[3]

(1. 西南交通大学 机械工程学院，成都 610031; 2. 成都工业学院 机械工程学院，成都 611730; 3. 成都航空职业技术学院 航空制造工程系，成都 610100)

摘　要：针对图像匹配问题进行了研究，提出了一种改进 BRIEF 算法的特征点匹配算法。该算法利用随机点与特征点之间的差分大小和差分幅值关系来生成特征点描述算子。针对 BRIEF 对噪声敏感问题，因为小的像素幅值差分更易受到噪声影响，为了抑制噪声，通过设置小像素差分阈值，差分在阈值内的设置为不确定位，然后通过其邻域均值来决定不确定位的值。特征点匹配使用描述算子之间的汉明距离进行比较来完成。实验与 BRIEF 和 ORB 算法进行了比较，证明该算子具有更高的判别性，计算简单且具有很好的噪声抑制性能，运行速度快，匹配准确率更高。

关键词：BRIEF；ORB；特征点匹配算子；局部二值模式

中图分类号：TP391.4　　doi: 10.3969/j.issn.1001-3695.2017.07.0890

Local binary pattern image feature point matching algorithm

Wang Qiang[1, 2], Li Bailin[1], Luo Jianqiao[1], Chen Xiaoyan[3]

(1. *School of Mechanical Engineering*, *Southwest Jiaotong University*, *Chengdu* 610031, *China*; 2. *College of Mechanical Engineering*, *Chengdu Technological University*, *Chengdu* 611730, *China*; 3. *Dept. of Aviation Manfacturing Engineering*, *Chengdu Aeronautic Polytechnic*, *Chengdu* 610100, *China*)

Abstract: This paper researched Image matching issue and proposed a feature point matching algorithm to improve the BRIEF algorithm. The proposed algorithm generated the feature point description operator according to the difference signs and the difference magnitudes relation between the random point and the feature point. BRIEF was sensitive to noise because the small difference magnitude is more susceptible to the noise. To solve this problem, This paper determined a small pixel difference threshold by the neighborhood mean value of the BRIEF. Comparing the Hamming distance between descriptions realizes the feature points matching. Compared with BRIEF and ORB algorithm, the experiments proved that the operator has higher discriminant, simple calculation and good noise suppression performance. And the matching accuracy is higher.

Key Words: BRIEF; ORB; feature point matching operator; local binary pattern

0　引言

计算机视觉技术已经广泛应用在目标检测[1]、人脸识别[2~4]、图像分类[5]、场景识别[6]、运动物体跟踪[7]、纹理分类[8~12]等方面。其中，利用图像特征点进行匹配是计算机视觉技术中比较常用的一种技术。随着智能设备的发展，图像匹配算法的要求主要体现在稳定性、抗噪声以及低内存消耗和高运算速度，因此对特征点匹配算子提出了更高的要求。

图像特征点匹配算子的唯一性是基于特征点的图像匹配的关键。为了让描述算子更加稳定，通常采用更高的维度和更复杂的描述方式。高维的描述算子可以增加其判别性，但是计算相对复杂，且时间复杂度较高。BRIEF (binary robust independent elementary features) [13]算法对特征点描述给出了一种简单的方法。在此之前，Moravec[14]提出了基于特征点的图像匹配方法，Harris 等人[15]提出了 Harris 角点检测方法应用于图像匹配，但是匹配效果不是很理想。SIFT[16]算法采用了一种复杂的描述方式，在特征点 16×16 邻域内，利用梯度方向直方图进行描述，分别计算特征点邻域每个 4×4 小块 8 个方向的梯度方向直方图，得到 128 维向量，虽然算法比较稳定，但是效率较低。利用 PCA[17]或 LDA[18]方法能够部分提高速度。SURF[19~21]算法对 SIFT 进行了改进，采用 Haar 小波检测图像在水平和垂直方向的响应得到 64 维向量对特征点进行描述，在速度上有一定提升。后来有人又提出了 BRISK[22]和 FREAK[23]算法，在速度上有很大的提升，但是匹配准确率不够。BRIEF 算法在邻域内选

基金项目：国家自然科学基金资助项目(51275431)；四川省科技厅项目(2016GZ0194)；四川省教育厅项目(16ZB0330)；四川省创新创业项目(201611116007)

作者简介：王强（1979-），男，四川广安人，副教授，博士研究生，主要研究方向为计算机视觉、模式识别（wqiangsky@163.com）；李柏林（1962-）男，教授，博导，主要研究方向为计算机视觉、模式识别；罗建桥（1991-），男，博士研究生，主要研究方向为计算机视觉、模式识别；陈小艳（1980-），女，硕士，助教，主要研究方向为计算机视觉、模式识别.

择随机点对，通过比较点对像素大小得到二进制串作为特征点描述算子，通过异或运算在时间上大大提高，但是单个像素之间比较对噪声敏感。ORB[24]算法对 BRIEF 进行了改进，以随机点为中心，取 $S \times S$ 子窗口像素之和进行比较得到二进制编码，能够部分抑制噪声。ORB 和 BRIEF 算法计算速度快。但是只考虑了特征点邻域内随机点的分布，而没有考虑邻域像素与特征点之间的关系。此外，BRIEF 算法容易受到噪声影响。ORB 算子通过均值滤波抑制了部分噪声，但是为了解决旋转不变性问题，计算特征点的主方向导致描述算子的相关性增加，从而判别性下降。本文提出利用随机点与特征点之间的关系来描述特征点的属性。此外，图像如果受到噪声影响，小的像素差分幅值影响更加明显，由此，本文根据差分幅值大小设置阈值单独处理这种情况，降低了噪声。

1 BRIEF 和 ORB 简介及问题分析

1.1 BRIEF 算法简介

BRIEF 算法用于特征点匹配，采用特征点邻域随机点对进行比较，得到二进制串，通过汉明距离判断两个特征点是否匹配。在特征点邻域内取 $S \times S$ 大小的邻域 p，定义测试函数 τ

$$\tau(p;x,y):=\begin{cases}1 & if\ p(x)\geq p(y)\\ 0 & otherwise\end{cases} \tag{1}$$

$p(x)$ 和 $p(y)$ 代表点 x 和点 y 位置的像素值。如果取 n_d 维二进制串进行特征点描述,则 BRIEF 的描述算子可以表示为

$$f_{n_d}(p):=\sum_{1\leq i\leq n_d} 2^{i-1}\tau(p;x_i,y_i) \tag{2}$$

其中：$n_d = 128, 256, 512$，代表二进制串的位数，特征点匹配采用汉明距离进行判断，最小的汉明距离认为是匹配的点对。随机点的选择服从高斯分布，X 和 Y 服从各向同性高斯分布 $(x,y)\sim Gaussian(0,\frac{1}{25}S^2)$。BIREF 算法速度很快，但是对噪声敏感，在得到描述算子之前需要先进行高斯滤波。

1.2 ORB 算法简介

ORB 算法对 BRIEF 进行了改进，主要是解决 BRIEF 算法对噪声敏感和旋转不变性问题。其点对选择方式和 BRIEF 一样。为了解决噪声敏感的问题，采用了图像积分的方法，本质上是对图像进行了均值滤波。为了解决旋转不变性问题，通过计算特征点的主方向，再根据主方向来生成描述算子。但是由于主方向会发生变化，随机点对的相关性会比较大，从而降低描述子的判别性。

1.3 问题分析

相比较其他基于特征点的匹配算法，BRIEF 和 ORB 具有速度快等优点。但是其缺陷也很明显，首先是噪声敏感问题不能很好地解决；其次特征点描述算子只考虑了邻域内随机点之间的差分大小关系，没有考虑差分幅值的关系，特征点与邻域的差分分布关系表达不够完整，所以得到的描述算子判别性不高。

2 提出的方法

2.1 理论分析

Ojala 等人[25~26]提出，图像的纹理特征可以通过中心像素与邻域像素的差分关系表示，由此提出了 LBP 编码算法。如果将中心像素看成是一个特征点，则 LBP 算法可以认为是关于特征点的描述算子。该描述算子利用了邻域像素和中心像素的差分分布关系。如果描述算子表达的差分关系足够完整，得到的描述算子将具有更高的判别性。邻域像素和中心像素的差分关系可以由差分大小和差分幅值完全描述。差分大小关系可以表达大部分原始信息，LBP 或 BRIEF 只利用了差分大小关系。但是差分幅值也可以表达部分原始信息[8]，显然这部分信息对提高描述算子的判别性也是有用的。设 $N \times N$ 邻域中心像素为 g_c，邻域像素为 $g_p(0,1,\cdots,p-1)$，定义 g_p 与 g_c 的差分 $z_p = g_p - g_c$，z_p 可以分解为差分大小和幅值两部分

$$z_p = b_p * m_p \tag{3}$$

其中：

$$b_p=\begin{cases}1 & z_p\geq 0\\ -1 & z_p<0\end{cases} \tag{4}$$

b_p 代表像素差分大小关系，m_p 代表像素差分幅值信息。Maitre[27]已经证明,像素之间的差分服从高斯分布或拉普拉斯分布:

$$f(x)=\exp(-|x|/\sigma)/\sigma \tag{5}$$

而像素差分大小关系 b_p 服从伯努利分布：

$$f(n)=a^{\frac{n+1}{2}}(1-a)^{\frac{1-n}{2}}。 \tag{6}$$

其中：$a = 0.5$，$n\in\{-1,1\}$。

如果分别从 b_p 和 m_p 重建差分信号 z_p，当只采用幅值信息重建差分信号 z_p，则

$$z_{pm}{}' = m_p * b_p{}' \tag{7}$$

$b_p{}'$ 服从伯努利分布，z_p 服从拉普拉斯分布,所以 m_p 将服从单边拉普拉斯分布:

$$f(x)=\exp(-x/\sigma)/\sigma, x>0。 \tag{8}$$

如果只使用像素差分大小关系重建原始信号，则

$$z_{pb}{}' = m_p{}' * b_p \tag{9}$$

$m_p{}'$ 表示幅值部分的均值。重建误差分别由 E_b 和 E_m 表示为

$$E_b = E[(z_p - z_{pb}{}')^2] \tag{10}$$

$$E_m = E[(z_p - z_{pm}{}')^2] \tag{11}$$

由此可以得到 $E_b = \sigma^2$ 而 $E_m = 4\sigma^2$ [8]。可以看出，信号之间的差分幅值信息能够表示原始图像 20%的信息。如果能够利用差分幅值信息，将大大提高 BRIEF 的判别性。

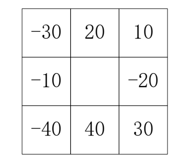
2.2 方法

为了更完整地表达特征点与邻域的差分关系，提高描述算子的判别性，本文采用差分大小和差分幅值联合生成特征点描述算子。同时，根据图像之间的差分服从高斯分布或拉普拉斯

分布的特点[27]，在特征点邻域内采用高斯分布生成随机点，利用随机点和特征点之间的关系来生成特征点的二进制描述算子。图 1 是整个特征点描述算子生成示意图。其中图 1(a)表示特征点附近 3×3 邻域；(b)表示邻域像素与中心像素的差分；(c)表示差分信号大小关系，其中“-1”在实际编码时用“0”代替；(d)表示差分幅值大小。

20	70	60
40	50	30
10	90	80

(a)特征点及邻域



−30	20	10
−10		−20
−40	40	30

(b)邻域与中心差分

−1	1	1
−1		−1
−1	1	1

(c)差分信号

30	20	10
10		20
40	40	30

(d)差分幅值

图 1　差分信号关系和差分幅值大小编码示意

设邻域大小为 $N\times N$，为了生成特征点描述算子，首先根据高斯分布，选取以特征点为中心的 $N\times N$ 邻域内 K 个点，设特征点像素值 f_c，记特征点与邻域内的像素值联合分布为

$$T=t(f_c,f_1,f_2,\cdots,f_k) \tag{12}$$

为了满足灰度不变性，让邻域像素减去中心像素的值，得到式(13)。

$$T=t(f_1-f_c,f_2-f_c,\cdots,f_k-f_c) \tag{13}$$

定义符号函数 $s(x)$：

$$s(x)=\begin{cases}1 & x\ge 0\\ 0 & x<0\end{cases} \tag{14}$$

其中：$s(x)=s(f_k-f_c)$，如果只考虑差分信号大小关系，则得到式(15)

$$T_s=t(s(f_1-f_c),s(f_2-f_c),\cdots,s(f_k-f_c)) \tag{15}$$

式（15）是邻域像素与特征点差分大小关系的纹理描述算子，具有很高的判别性[25~26]，但是没有差分幅值信息，导致丢失了部分邻域像素和特征点之间的关联信息。为了保证信息不丢失，再进行差分幅值编码。由于幅值大小不能够直接编码成“0”或“1”，所以设置每个邻域像素与特征点差分幅值之和的均值作为阈值，将每个幅值大小与均值进行比较得到“0”或“1”的编码，得到式(16)。

$$T_m=t(s(m_1-m_m),s(m_2-m_m),...,s(m_k-m_m)) \tag{16}$$

其中：m_k 表示邻域像素与特征点的幅值大小；m_m 表示幅值的均值；T_m 即幅值的二进制编码结果，将 T_s 和 T_m 联合即得到特征点的二进制描述算子。

2.3　噪声抑制

由于单个像素进行比较容易受到噪声影响，为了抑制噪声，采用局部邻域均值代替特征点作为阈值，从而消除部分噪声，设邻域像素均值为 f_m。

$$f_m=\frac{1}{k+1}(\sum_{i=1}^{k}f_i+f_c) \tag{17}$$

计算特征点邻域像素与均值的差分。

$$T=t(f_1-f_m,f_2-f_m,\cdots,f_k-f_m) \tag{18}$$

采用邻域均值作为阈值进行差分计算，可以抑制部分噪声。文献[2]提出小的像素差分幅值更容易受到噪声影响。为了进一步抑制噪声，设定阈值 t，将邻域像素与均值的差分幅值在阈值 t 范围内的编码设置为不确定位 X。由此修改符号函数为

$$s'(x)=\begin{cases}1 & x>t\\ X & |x|\le t\\ 0 & x<-t\end{cases} \tag{19}$$

不确定位 X 取“0”或“1”，具体取值由该像素的邻域决定。设点 f_p 为不确定点，取 f_p 点附近 3×3 邻域共 9 个点 f_{pi}，计算平均值 f_{pm}，由 f_p 的邻域点共同决定 f_p 的值。

$$f_{pm}=\frac{1}{9}\sum_{i=1}^{9}f_{pi} \tag{20}$$

用 f_{pm} 代替 f_p，然后再与 f_m 进行比较得到 X 的值。相当于进行了局部均值滤波。由此消除小像素差异的噪声，得到差分大小关系的二进制编码结果，如式(21)所示。

$$f_{kd}(p):=\sum_{1\le i\le kd}2^{i-1}s'(x) \tag{21}$$

其中：kd 代表 $N\times N$ 邻域的 k 维点。

同理可以得到差分幅值的二进制编码结果。将两个编码结果进行线性连接得到整个抗噪声的二进制编码。

采用高斯分布选择邻域内随机点与特征点进行比较，并且不仅仅利用差分大小信号，同时利用了差分幅值信号，从而更加准确地描述了特征点的属性。将邻域像素的均值作为阈值进行比较，排除了部分噪声影响，对于小的像素幅值差异引起的噪声，采用不确定位的编码方式，通过计算 3×3 邻域均值代替邻域像素的值与均值进行比较，进一步提高了描述算子的抗噪声能力。其算法描述如下：

提出的特征点匹配算子

1. 利用高斯函数生成邻域内随机点的位置；

2. for 循环处理每个特征点；

2.1.计算特征点 $N\times N$ 邻域的均值 f_m；

2.2.设置阈值 t，比较随机点 f_p 与均值 f_m 的大小

if　$f_p-f_m>t$　then　bitset=1

else if　$f_p-f_m<-t$　then　bitset=0

else if　$|f_p-f_m|\le t$　then

计算 f_p 的 3×3 邻域均值 f_{pm}

if　$f_{pm}-f_m\ge 0$　then　bitset=1

else　bitset=0

得到关于差分信号大小的二进制编码 T_s；

2.3. 计算每个随机点与特征点的幅值 m_i；

2.4 设置阈值 t，计算幅值的均值 m_m，比较 m_i 与 m_m 大小，与步骤 2.2 类似，得到关于幅值的二进制编码 T_m；

3. 将 T_s 与 T_m 结合，得到特征点的描述算子。

3　实验结果

为了验证提出的特征点匹配算法，实验采用标准的图像匹配基准数据集进行测试。实验比较了 BRIEF 和 ORB 算法。图 2 是五个公开可用的测试图像序列。每个图像序列 6 张图像，分别表示视觉变化(wall)、图像模糊(trees) 、光照变化(light)、压缩图像(JPG)和旋转加尺度变换(boat)。小像素差异阈值的选择与噪声的强弱有关，如果噪声比较小，阈值选择通常在 2~5。如果噪声很强，阈值相应增大，这样会增加部分时间开销，如果图像本身的噪声很强，在进行特征点检测阶段会进行降噪声的预处理，所以特征点描述阶段的阈值一般 2~5 范围是比较合适的。实验中选择的阈值为 $t=5$。

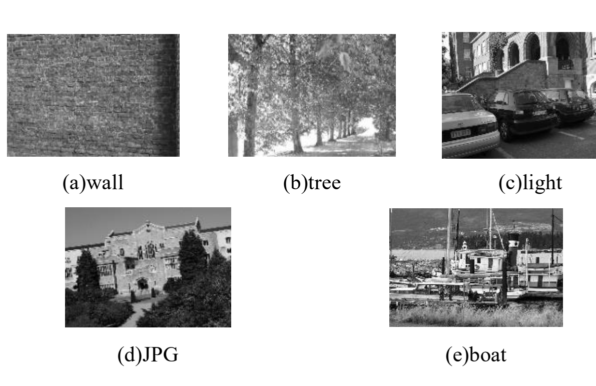

(a)wall　(b)tree　(c)light
(d)JPG　(e)boat

图 2　测试图像集

图 3 是选择不同邻域大小的匹配结果示意。以图 2(b)图像序列 6 张图像为例，第一张图像分别与后面五张图像匹配。N 表示以特征点为中心的邻域大小。通常邻域选择越大，选择的随机点越多，匹配越准确，N=17 以后，邻域增加对匹配率的结果影响不是很大，而且邻域越大计算时间越多，对于匹配要求比较高的情况下通常选择 N=17 的邻域已经足够，如果想进一步提高时间，可以选择更小的邻域，但是有可能导致匹配率下降。

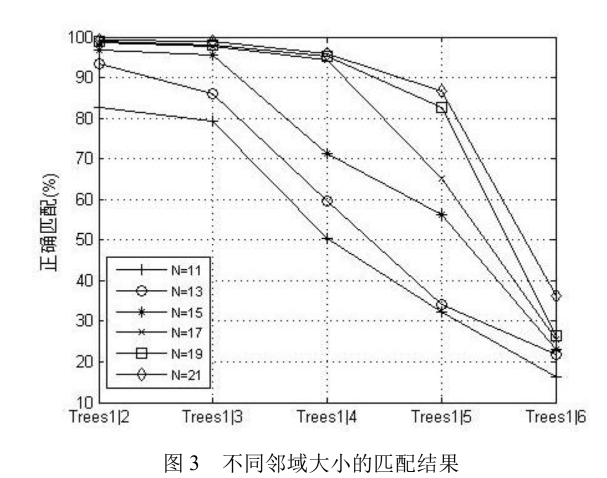


图 3　不同邻域大小的匹配结果

实验从两个方面进行比较：同一序列图像分别与第一张图像分别进行匹配；对每个序列第二张图像分别添加不同的高斯噪声和椒盐噪声再与第一张图像进行匹配。实验采用最直观的指标衡量匹配结果，即匹配准确率。实验编码采用 c++结合 opencv2.4.9 实现。邻域大小 N=17，小像素差异阈值选择 $t=5$，特征点采用 CenSurE 特征点。表 1~5 显示了原始图像序列第一张图像分别与剩下的五张图像的匹配结果。其中 BRIEF 和 ORB 算法选择的是 512 位的描述算子。从图中可以看出，对于光照、视觉、模糊和压缩变化的图像，本文提出的算法匹配率最高，对于旋转变化和尺度变化的图像，如果旋转和尺度变化不大，本文算法要好于 ORB 算法，如果增大旋转角度和尺度，本文算法略低于 ORB 算法。由于 ORB 算法增加了主方向方向的计算，所以旋转不变性较好。但是增加主方向会导致随机点对选择相关性比较大，从而描述算子的判别性有所下降，这一点从表 1~4 可以看出。现实中很多图像旋转和尺度变化不会太大，所以本文算法是适用的。

表 1　Wall 图像匹配率/%

方法	1|2	1|3	1|4	1|5	1|6
ORB	93.8	92.5	90.3	70.7	57.6
BRIEF	95.1	93.4	93.1	75.8	60.9
本文方法	95.4	94.8	94.4	77.2	62.6

表 2　Trees 图像匹配率/%

方法	1|2	1|3	1|4	1|5	1|6
ORB	92.2	90.7	88.2	73.4	57.1
BRIEF	95.1	93.1	91.5	75.8	62.4
本文方法	96.7	94.5	92.6	76.3	64.9

表 3　Light 图像匹配率/%

方法	1|2	1|3	1|4	1|5	1|6
ORB	92.1	90.3	89.8	88.7	85.7
BRIEF	94.2	92.5	91.4	91.3	87.6
本文方法	94.5	92.6	92.3	92.5	88.3

表 4　JPG 图像匹配率/%

方法	1|2	1|3	1|4	1|5	1|6
ORB	92.7	90.3	88.2	73.9	57.2
BRIEF	96.1	93.3	91.7	75.4	62.9
本文方法	98.8	94.7	92.5	76.8	64.3

表 5　Boat 图像匹配率/%

方法	1|2	1|3	1|4	1|5	1|6
ORB	91.6	90.8	86.1	70.3	55.8
BRIEF	94.3	88.3	84.9	65.9	52.1
本文方法	95.2	89.5	85.6	68.5	53.7

图 4 是添加了高斯噪声的图像示意，选择 trees 图像作为示例。高斯噪声是一种最常见的噪声。根据标准偏差 σ 的大小，选择了四种 σ 值进行实验，分别是 $\sigma=0.05$，0.10，0.15，0.2。随着 σ 值的增大，图像越模糊，匹配的准确率越低。图 5 是添加椒盐噪声的图像结果，同样选择 Trees 图像作为示例。椒盐

噪声是另一种比较常见的图像噪声。选择的椒盐噪声的大小分别是 p=0.05，0.10，0.15，0.20。与高斯噪声类似，当图像噪声增大后，匹配率也相应降低，当噪声很大时，用肉眼已经无法观察图像的细节特征。

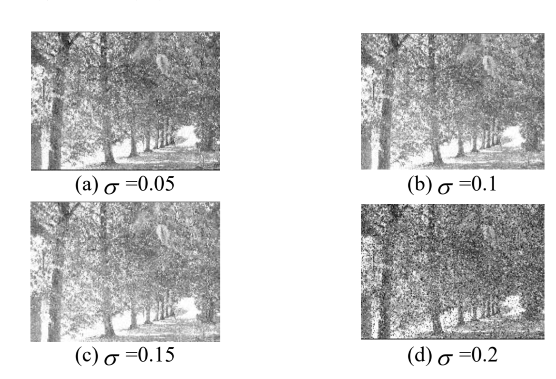


图 4 添加高斯噪声的图像

表 6~10 是添加了高斯噪声后的匹配结果。表 11~15 是添加了椒盐噪声后的匹配结果。随着噪声的增加，匹配率在下降，但是不管添加哪种噪声，本文算法都要好于 BRIEF 和 ORB 算法。BRIEF 算法在计算描述算子之前进行了高斯滤波，所以有较强的噪声抑制能力，其匹配效果要好于 ORB 算法。而 ORB 算法采用局部邻域像素之和进行描述算子计算。本质上是利用均值滤波代替了高斯滤波，通过图像积分提高了算子的计算速度，同时有一定的噪声抑制能够，但是主方向的计算导致随机点对相关性增加从而导致描述算子判别性下降的问题没有很好解决，从而匹配率有所下降。

表 6 Wall 图像添加高斯噪声匹配率/%

方法	$\sigma=0.05$	$\sigma=0.10$	$\sigma=0.15$	$\sigma=0.20$
ORB	92.1	85.7	80.3	73.6
BRIEF	93.5	91.7	90.5	88.3
本文方法	94.3	93.5	92.8	89.4

表 7 Trees 图像高斯噪声 $\sigma=0.10$ 匹配率/%

方法	$\sigma=0.05$	$\sigma=0.10$	$\sigma=0.15$	$\sigma=0.20$
ORB	93.4	86.9	81.8	76.5
BRIEF	95.3	90.2	87.4	84.8
本文方法	95.7	91.6	89.3	87.3

表 8 Light 图像高斯噪声 $\sigma=0.15$ 匹配率/%

方法	$\sigma=0.05$	$\sigma=0.10$	$\sigma=0.15$	$\sigma=0.20$
ORB	93.5	90.2	89.8	87.3
BRIEF	94.3	91.1	91.4	89.9
本文方法	95.8	92.6	91.6	90.3

表 9 JPG 图像高斯噪声 $\sigma=0.20$ 匹配率/%

方法	$\sigma=0.05$	$\sigma=0.10$	$\sigma=0.15$	$\sigma=0.20$
ORB	95.4	94.1	93.7	91.2
BRIEF	96.1	95.7	95.2	94.2
本文方法	97.7	96.4	95.5	95.1

表 10 Boat 图像高斯噪声 $\sigma=0.20$ 匹配率/%

方法	$\sigma=0.05$	$\sigma=0.10$	$\sigma=0.15$	$\sigma=0.20$
ORB	94.5	91.8	83.6	72.4
BRIEF	95.3	93.6	85.8	79.7
本文方法	96.6	94.9	86.1	81.5

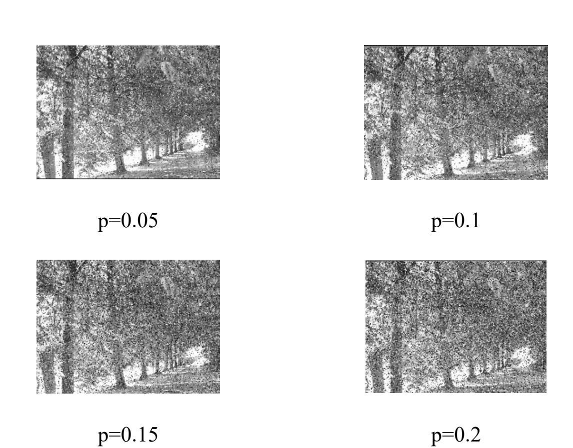


图 5 添加椒盐噪声的图

表 11 Wall 图像匹配率/% p=0.05

方法	p=0.05	p=0.10	p=0.15	p=0.20
ORB	92.2	88.9	70.7	53.3
BRIEF	94.6	92.7	85.3	78.5
本文方法	95.7	94.3	87.6	80.1

表 12 Trees 图像匹配率/% p=0.10

方法	p=0.05	p=0.10	p=0.15	p=0.20
ORB	91.7	88.5	70.2	53.5
BRIEF	93.7	92.7	85.7	78.1
本文方法	95.4	94.5	87.2	80.8

表 13 Light 图像匹配率/% p=0.15

方法	p=0.05	p=0.10	p=0.15	p=0.20
ORB	90.7	89.1	82.7	62.4
BRIEF	92.7	90.2	85.5	74.7
本文方法	93.4	91.6	87.6	78.7

表 14 JPG 图像匹配率/% p=0.20

方法	p=0.05	p=0.10	p=0.15	p=0.20
ORB	92.1	90.4	73.6	61.7
BRIEF	95.6	93.3	84.5	80.3
本文方法	98.3	97.7	84.8	82.7

表 15 Boat 图像匹配率/% p=0.20

方法	p=0.05	p=0.10	p=0.15	p=0.20
ORB	93.7	90.3	83.1	63.5
BRIEF	95.1	92.9	85.1	75.2
本文方法	95.8	94.1	86.4	77.7

从实验结果可以看出，不管是原始图像之间的匹配还是添加了噪声的图像进行匹配，改进的方法都能够得到更好的匹配

结果。由于改进的方法增加了差分幅值信息，得到的描述算子能够完整的表达邻域像素与特征点之间的联合差分分布关系，其判别性更强。此外，通过设置小像素差异阈值，提高了算法的抗噪声能力。

从描述算子的运行时间来看，由于都是采用的汉明距离进行判断，三种算法进行匹配的时间复杂度接近。时间开销主要在描述算子的生成上面。BRIE 进行高斯滤波操作将花费大量的时间，如果不进行高斯滤波其速度将是最快的，但是缺少滤波操作将大大降低匹配准确率。ORB 算法利用图像积分操作代替高斯滤波，时间要小于 BRIEF。本文算法在生成描述算子的时候，时间主要消耗在进行小像素差值阈值比较阶段，图像上的小像素差值的数量根据具体的图像有所不同，其时间与需要进行邻域均值计算的像素点的数量有关。但是该方法同样可以利用图像积分快速实现，因此，时间复杂度与 ORB 接近，唯一不同的是因为利用了差分信号和差分幅值两种关系得到描述算子，本文的描述算子长度要比 BRIEF 或 ORB 多一倍，所以整体上时间略多于 ORB 算法，最后的实验也证明了这一点。设有 n 个特征点，在计算描述算子时只有一次循环，计算次数为 n，因此时间复杂度为 $O(n)$。从整体上比较，在同时采用 512 位二进制串进行比较的情况下，本文算法与 ORB 计算时间接近，比 BRIEF 要快。表 16 是三种算法计算描述算子的运行时间比较。算法在单线程 Intel i5 2.5 GHz 处理器上实现。特征点数量为 875 个点。时间采用多次运行的平均结果。

表 16　运行时间比较

方法	ORB	BRIEF	本文方法
时间/ms	70.9466	171.545	83.1741

4　结束语

BREIF 算法是比较优秀的算法，在保证匹配率的前提下，其运行效率非常高，ORB 算法在 BRIEF 算法上进行了改进，增加了旋转不变性，但是导致描述算子的相关性增加。从而降低了匹配率，但是增加了旋转不变性。这两个算法都对噪声敏感，所以 BRIEF 算法在使用前需要进行高斯滤波。ORB 算法采用图像积分来实现滤波操作，时间上有了很大提高。但是 BRIEF 算法和 ORB 算法都只是考虑了特征点邻域之间的分布关系，没有考虑特征点和邻域之间的关系，同时，只考虑了像素差分大小关系，没有考虑像素差分幅值关系。本文从像素差分大小和幅值关系两方面进行考虑得到二进制描述算子，能够更完整的表达邻域之间的联合差分分布关系，得到的描述算子判别性更高。通过邻域均值以及小像素差异之间设置阈值的方式进行噪声抑制，避免了高斯滤波操作。描述算子采用二进制串之间的异或操作计算匹配距离，计算简单，速度快。

参考文献：

[1] Satpathy A, Jiang X, Eng H L. LBP-based edge-texture features for object recognition [J]. IEEE Trans on image Processing, 2014, 23 (5): 1953-1964.

[2] Ren J, Jiang X, Yuan J, et al. LBP encoding schemes jointly utilizing the information of current bit and other LBP bits [J]. IEEE Signal Processing Letters, 2015, 22 (12): 2373-2377.

[3] Chen D, Cao X, Wen F, et al. Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2013: 3025-3032.

[4] Deshmukh A, Sirpotdar P, Sheikh J, et al. High accuracy face recognition system based on SIFT [J]. International Journal of Advanced Research in Computer Engineering & Technology, 2015, 4 (6): 2626-2628.

[5] Li W, Chen C, Su H, et al. Local binary patterns and extreme learning machine for hyperspectral imagery classification [J]. IEEE Trans on Geoscience and Remote Sensing, 2015, 53 (7): 3681-3693.

[6] Xiao Y, Wu J, Yuan J, et al. mCENTRIST: a multi-channel feature generation mechanism for scene categorization [J]. IEEE Trans on Image Processing, 2014, 23 (2): 823-836.

[7] Zhao G, Ahonen T, Matas J, et al. Rotation-invariant image and video description with local binary pattern features [J]. IEEE Trans on Image Processing, 2012, 21 (4): 1465-1477.

[8] Guo Z, Zhang L, Zhang D, et al. A completed modeling of local binary pattern operator for texture classification [J]. IEEE Trans on Image Processing, 2010, 19 (6): 1657-1663.

[9] Song T, Li H, Meng F, et al. Noise-robust texture description using local contrast patterns via global measures [J]. IEEE Signal Process. Letters, 2014, 21 (1): 93-96.

[10] Wang K, Bichot C E, Zhu C, et al. Pixel to patch sampling structure and local neighboring intensity relationship patterns for texture classification [J]. IEEE Signal Processing Letters, 2013, 20 (9): 853-856.

[11] Liao S, Law M W K, Chung A C S, et al. Dominant local binary patterns for texture classification [J]. IEEE Trans on Image Processing, 2009, 18 (5): 1107-1118.

[12] Kwak J T, Xu S, Wood B J, et al. Efficient data mining for local binary pattern in texture image analysis [J]. Expert Systems with Applications, 2015, 42 (9): 4529–4539.

[13] Calonder M, Lepetit V, Ozuysal M, et al. BRIEF: computing a local binary descriptor very fast [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2012, 34 (7): 1281-1298.

[14] Moravec H P. Rover visual obstacle avoidance [C]// Proc of the 7th International Joint Conference on Artificial Intelligence. 1981: 785-790.

[15] Harris C, Stephens M. A combined corner and edge detector [C]// Proc of the 4th Alvey Vision Conference. 1988: 147-151.

[16] Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91-110.

[17] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2005, 27 (10): 1615-1630.

[18] Hua G, Brown M, Winder S. Discriminant embedding for local image descriptors [C]// Proc of the 11th IEEE International Conference on Computer Vision. 2007: 1-8.

[19] Bay H, Tuytelaars T, Van Gool L. Surf: speeded up robust features [C]// Proc of International Conference on Computer Vision. Berlin: Springer, 2006: 404-417.

[20] 初守艳, 席志红. 结合 SURF 的数字稳像技术 [J]. 计算机辅助设计与图形学学报, 2014, 26 (2): 241-247.

[21] 陈小丹, 杜宇人, 高秀斌, 等. 一种基于 SURF 的图像特征点快速匹配算法 [J]. 扬州大学学报: 自然科学版, 2012, 15 (4): 64-67.

[22] Leutenegger S, Chli M, Siegwart R Y. BRISK: binary robust invariant scalable keypoints [C]// Proc of International Conference on Computer Vision. 2011: 2548-2555.

[23] Alahi A, Ortiz R, Vandergheynst P. Freak: fast retina keypoint [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2012: 510-517.

[24] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or URF [C]// Proc of International Conference on Computer Vision. 2011: 2564-2571.

[25] Ojala T, Pietikainen M, Harwood D. A comparative study of texture measures with classification based on featured distributions [J]. Pattern Recognition, 1996, 29 (1): 51-59.

[26] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24 (7): 971-987.

[27] Weinberger M J, Rissanen J J, Arps R B, et al. Applications of universal context modeling to lossless compression of gray-scale images [J]. IEEE Trans on Image Processing, 1996, 5 (4): 575-586.